ReRAM 1T1R Simulation complications

Abstract:
ReRAM has many potential uses as a memory one of which being used as a way to perform in memory computations. This can be done by creating a matrix of cells and applying voltage vectors across them to create an output. To do this each cell needs to consist of one ReRAM cell and one transistor. These cells will be called 1T1R cells. An example of one of these cells is shown in Figure 1. It is important to be able to model these cells when creating a crossbar array for in memory computation reasons. The Skywater130 nm PDK has open source ReRAM models for public use. However, because of the way that the Ngspice models behave it may be impossible to simulate the ReRAM properly. This document will cover the proper way to manipulate the ReRAM and how the models limit proper simulation function.

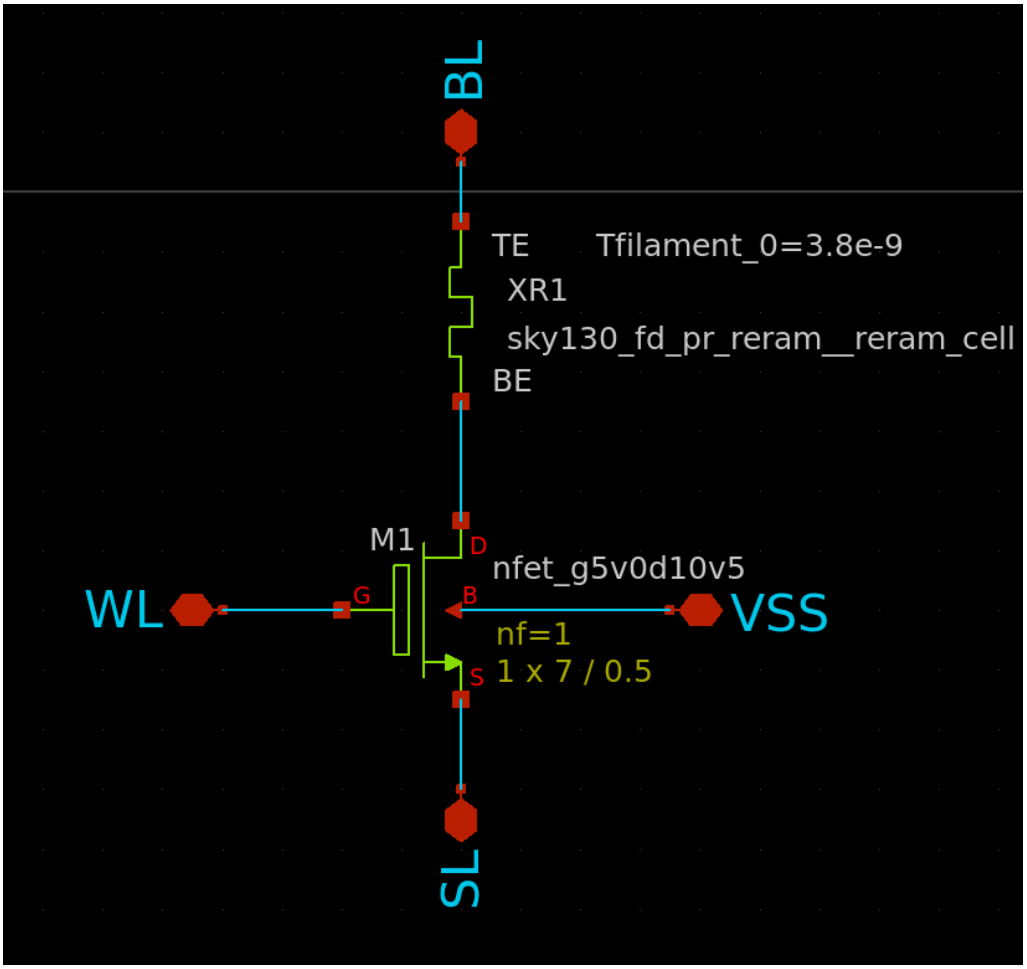

Figure 1 *1T1R Cell*

The operation of ReRAM can be explained by referring back to Figure 1. All ReRAM cells start in a pristine state and are unable to be used until they are formed. Forming is the process of creating conductive filaments through the ReRAM. This can be done by applying 3V at the bitline(BL) , turning on the transistor through the worldline(WL) and grounding the sourceline(SL). This puts the cell into a digital logic state of “1”, to get it to be a digital logic “0” 2.5V are applied at the sourceline while the bitline is grounded. To go from a zero to a one 2.5V are applied at the bitline and the sourceline is grounded. These operations are the simple controls to manipulating the ReRAM to take on the value that is needed.

Shown in Figures 2 and 3 is the current output of forming and resetting the 1T1R cell shown in Figure 1. It can be seen in Figure 3 that the cell does not reset when 2.5V is applied at the source line(yellow line in Figure 2). This can be seen because the current levels are equal before and after the reset operation, around 3.0us and after 6.0us respectively in Figure 3. This is because the transistor is hard coded to work only from source to drain and is not bidirectional. This problem can be circumvented by applying a negative 2.5V at the bitline instead, the results of which are shown in Figures 4 and 5. This works well for a single cell, however when the crossbar area is increased to a two by two matrix we can see that this causes issues as even though the worldline for a row can be off, the negative 2.5V will be applied to all cells evenly resulting in a cell that behave more like a “0” than the “1” it is thought to be. This is visualized in Figure 6 where is can be seen that even if WL1 is off the negative voltage at BL1 will still have an unwanted effect of the cell XR1.

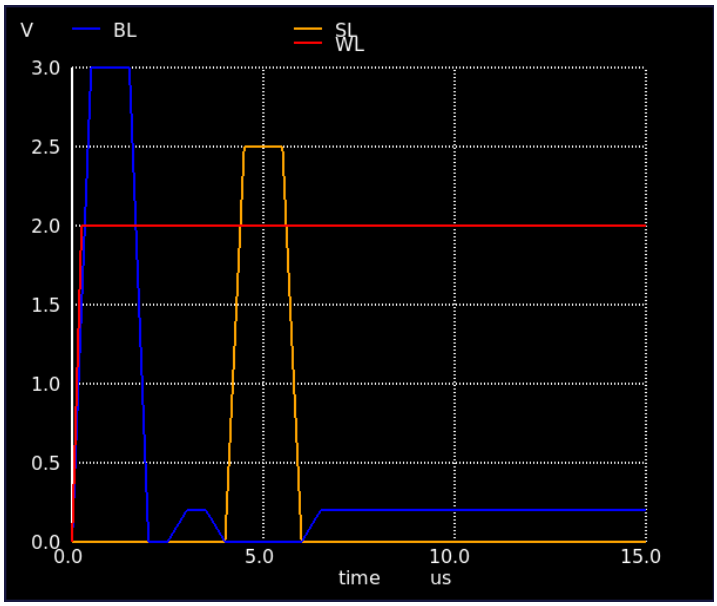

Figure 2 *Form and reset of 1T1R voltages*

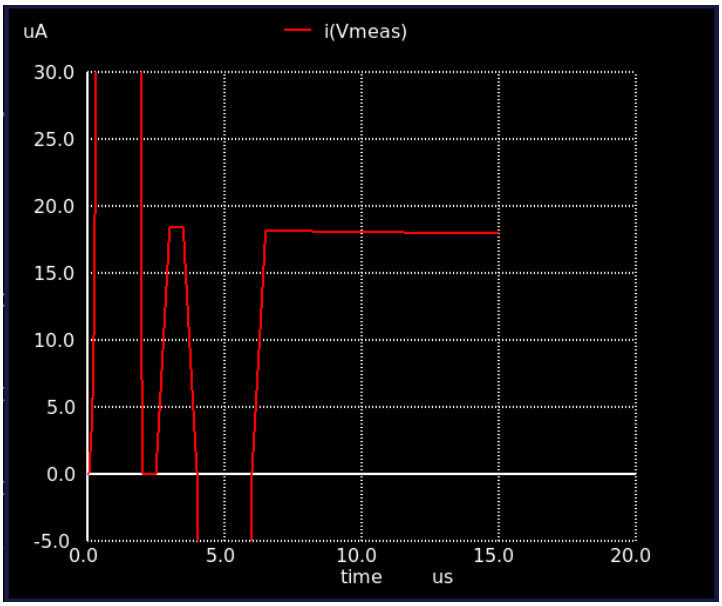

Figure 3 *Current from 1T1R form and reset*

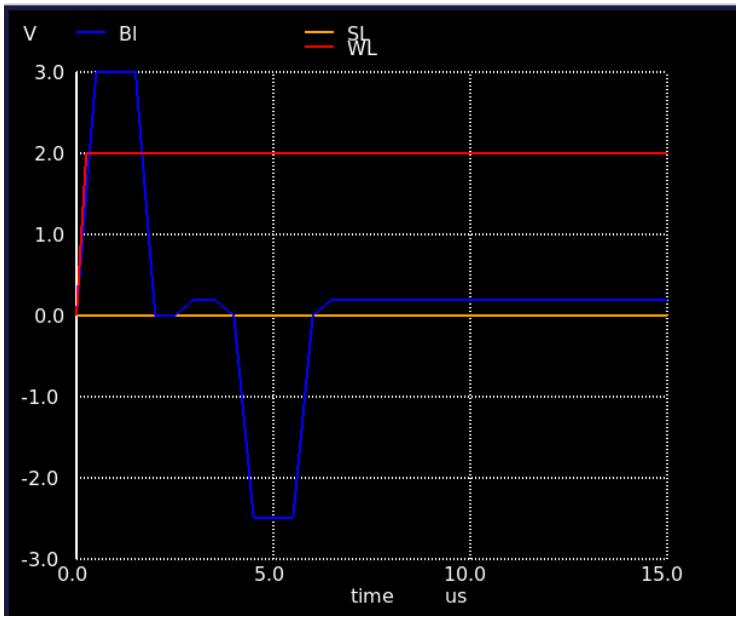

Figure 4 *Negative Bitline Voltage*

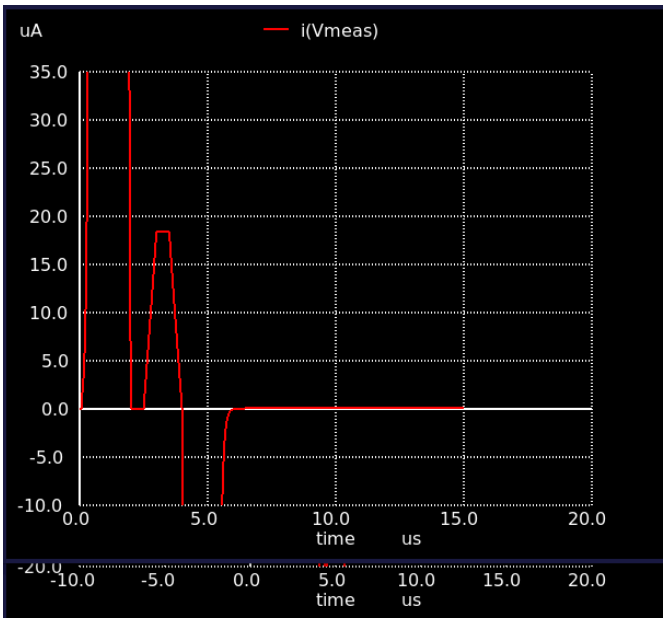

Figure 5 *Correct Current Output*

Proper simulation of a single 1T1R cell can be achieved by replacing the transistor as a resistor and allowing voltage to flow from both directions. This is useful for verifying that the model will work. However, we will still be unable to simulate a 2 by 2 or larger array as we need to be able to turn cells on and off so that we can set their values individually. This means that at the time of writing it is not possible to fully simulate a ReRAM crossbar but in theory, it will function correctly based on the output of a single cell. The cell operation will need to be verified on chip as well to make sure that the simulated results of a single cell match real world functionality.

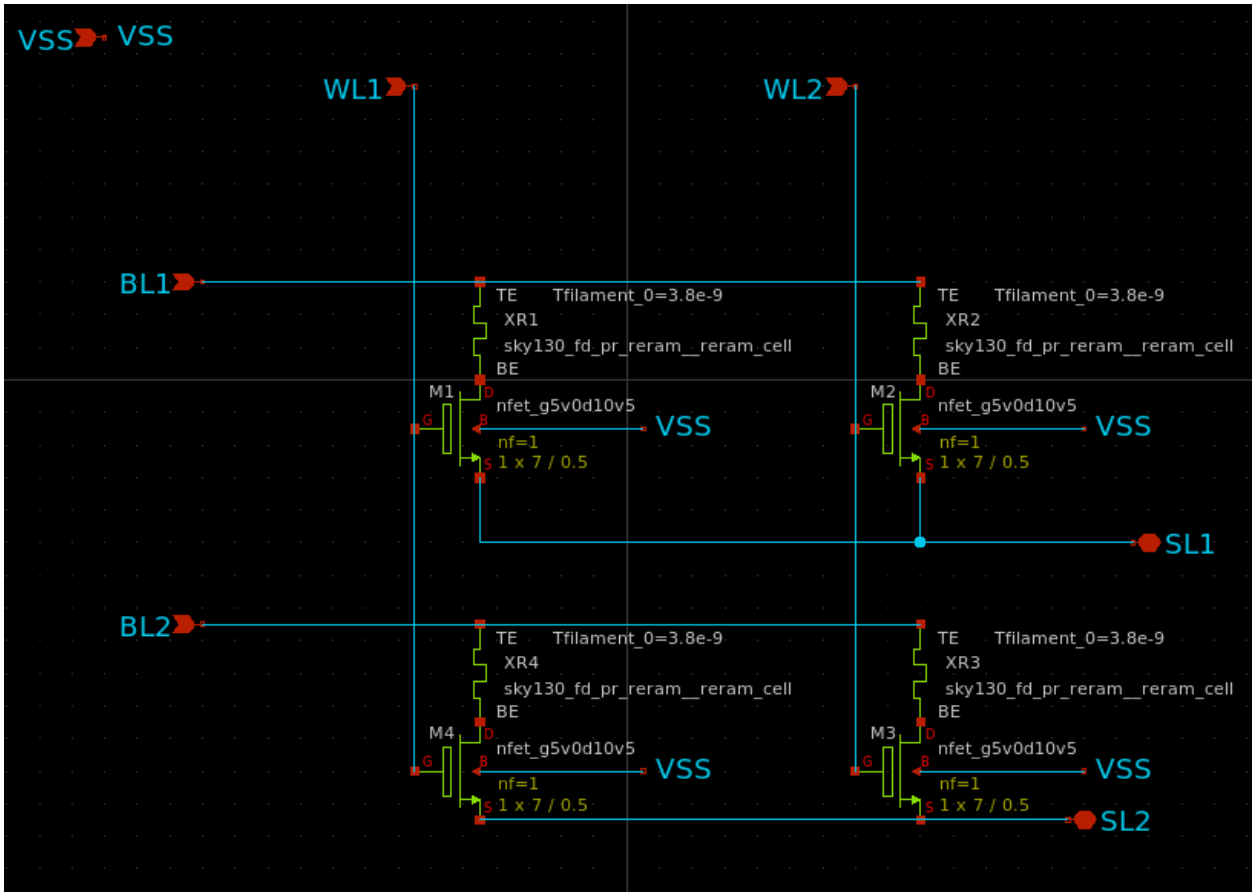

Figure 6 *2x2 ReRAM array*

Conclusion:

Modeling the ReRAM using the Skywater PDK has not proven to be as easy. This is due to the transistors in Ngspice being one direction from drain to source and not allowing voltage to flow source to drain. This creates a huge problem for simulating the ReRAM as to set and reset the device voltage needs to flow both ways. Being unable to properly simulate the device causes concern that the device may not work properly in actual fabrication, but in reality transistors do not know the difference between their own source and drain and will work bidirectionally.